Product Search Relevance

Manish Kumar
Computer Science
Indiana University
Bloomington,
Email: kumar20@iu.edu

Prateek Bhat
Computer Science
Indiana University
Bloomington
Email: pratbhat@iu.edu

Ritesh Agarwal
Data Science
Indiana University
Bloomington
Email: riteagar@iu.edu

***Abstract*—Search relevancy is a major part of big corporations like Google and Amazon. A lot of time has been spent on research to improve the recommendation systems. Shoppers rely on search relevancy to find the latest products and to get timely solutions to their needs. In this project we try to improve the relevancy score of a search query in order to suggest relevant products to users and improve the user experience for Home Depot website.**

I. Introduction

Shoppers rely on search relevancy to find the latest products and to get timely solutions to their needs. From installing a new ceiling fan to remodeling an entire kitchen, with the click of a mouse or tap of the screen, customers expect the correct results to their queries quickly. Speed, accuracy and delivering a frictionless customer experience are essential.

In this project, we will improve customers' shopping experience by developing a model that can accurately predict the relevance of search results.

Search relevancy is an implicit measure that is used to gauge how quickly customers can get to the right products. Currently, human raters evaluate the impact of potential changes to their search algorithms, which is a slow and subjective process. By removing or minimizing human input in search relevance evaluation, we hope to increase the number of iterations on the current search algorithms.

II. Background

Search relevancy is the practice of turning a search engine into a helpful sales associate. Most of the time users are not able to search what they are looking for based on the search query. Most e-commerce sites have been trying to deal with this issue since the beginning of time but it still exists as an open problem. Companies like Amazon, Netflix etc. have been trying to recommend relevant products to their users based on their use of different products.

The trick to relevancy is that search engines are simply sophisticated text matching systems. They can tell you when the search word matches a word in the document but they are not nearly as smart or adaptable like a human sales associate. Once a match is determined a search engine can use statistics about the relative frequency of that word to give a search result a relevancy score. Outside of this core engine a lot of search relevancy is about the development required to either jury-rig text to allow fuzzy matching or correctly boosting/weighting on the right factors. A developer working on search relevancy focuses on the following areas as the first line of defense:

- **Text Analysis**: The act of normalizing text from both a search query and a search result to allow fuzzy matching. For example, one step known as stemming can turn many forms of the same word shopped, shopping, and shopper all to a more normal form shop to allow all forms to match. Some extremely common words such as to,a,is,an which would appear to be of little value in helping select results needed by the user are excluded from the vocabulary entirely. These words are called stop words .

- **Query Time Weights and Boosts**: Re-weighting the importance of various fields based on search requirements. For example deciding a title field is more important than other fields.
- **Phrase/Position Matching**: Requiring or boosting on the appearance of the entire query or parts of a query as a phrase or based on the position of the words

The problem is interesting in the sense that we are not just trying to find relevancy of the search query based on similar terms but by the similar kind of product the user may be interested in, even when the there are no similar terms between the products.

III. Previous Work

Recommendation systems have always been a major part of big corporations like Google, Amazon etc. and there has been many ways of improving the same. Google uses Page ranking algorithm, which makes a web page relevant based on the number of hits generated on the page for a certain search query. Amazon has item to item collaborative filtering which recommends products for users based on the same items bought by other users. YouTube suggests videos based on viewed videos where more weight is given to the current videos being viewed by the user while less weight is given to the videos watched earlier, which means weight increases exponentially with videos watched in chronological time. Since these companies have large amount of data it is not possible to label search queries or similar search queries for most products and hence the constant need for evolving the algorithms.

IV. Acquiring Data

Kaggle is hosting a competition called Home Depot Product Search Relevance which is sponsored by Home Depot. The data contains information about the products like the product description, product attributes and real customer search queries from Home Depot's website. The challenge is to predict a relevance score for the provided combinations of search terms and products. To create the ground truth labels, Home Depot has crowd sourced the search/product pairs to multiple human raters.

The relevance is a number between 1 (not relevant) to 3 (highly relevant). For example, a search for ”AA battery” would be considered highly relevant to a pack of size AA batteries (relevance = 3), mildly relevant to a cordless drill battery (relevance = 2), and not relevant to a snow shovel (relevance = 1). Each pair has been evaluated by at least three human raters. The provided relevance scores are the average value of the ratings. In order to make sure that the relevancy scores are unbiased raters did not have access to the attributes and raters had access to product images, while the competition does not include images. The data that we have obtained contains the below fields:

- ID : a unique Id field which represents a (Search Term, Product ID) pair.
- Product ID : an Id for the products.
- Product Title : the product title.
- Product Description : Text description of the product (may contain HTML content).
- Search Term : Search query by users.
- Relevance - Average of the relevance ratings for a given Id.
- Name - Attribute name.

- Value - Attributes value.

V. Methodology

The first step in every Data Mining or Machine Learning algorithm is to clean the data available in hand, select the features in order to avoid the curse of dimensionality and creation of new features. After all this pre-processing of data an engineer is in a position to use any Machine learning or Data mining algorithm. Below are the methods that we have used for this project:

A. Spell Check

The search query in the data provided is what real users have entered in Home Depot's website to search for a product. We found that many search queries have typos for example a user was searching for ”portable fireplace” but entered it as ”potable fireplace”. There are many modules available for spell check in python but we have used Google search. We send ”search query” from our data to Google search which returns us the rectified term.

B. Stemming

For grammatical reasons, documents are going to use different forms of a word, such as notify, noticeable, notification and notation. Additionally, there are families of words with similar meanings, such as different, distinct, and divergent. In order to solve this problem we stem 'search query', 'product title', 'product description' and 'attributes' in the given data.

C. Feature Creation

Our goal is to get the best possibles results from our predictive model which have been described later, with the limited features of the data. The features in our training data will directly influence the results of our predictive model. Better the features simpler the model and better the results. We have created new features like common words in search query - product description and common words in search query - product title, length of search query, length of product title, length of product description, length of brand, ratio of common word in title to the length of search query, ratio of common words in description to length of search query.

This gives an idea about how relevant is the search query to that particular product. The number of common words can be considered as the weights given to the search query which will help us to predict the relevancy scores for that product.

D. Predictive Models

We have used Decision Tree and Random Forest as predictive models for our project. These supervised learning models are used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Decision tree have an advantage over other classifiers as they are easy to visualize, simple to interpret and understand. Also, the cost of using the tree is logarithmic in the number of data points used to train the tree. Though there are disadvantages to using decision tree like it can quiet easily over fit the model on the data which can be prevented by controlling the max depth of the tree or by pruning methods.
On the other hand a random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

E. Evaluation of a Model

Kaggle uses Root Mean Square (RMS) as a strategy to evaluate the results submitted by us. The results submitted are in a csv file which has a relevancy score for its respective id.

VI. Results

At first we tried to give an average score of 1.5 to all the product id's which served as a base line for our comparisons. The RMS value for the average score was 1.0287. To improve on the baseline, we tried different approaches.

First of all, we started with simple vectorization. Idea was to create bag of words and then see the feature vector with corresponding product. On the basis of the similarity, we assigned relevancy score to the search term, result obtained was not satisfactory. To improve on it, we applied the term vector on decision tree. Result obtained was a marked improvement on the previous scores obtained. Since, in this step we obtained the result using single decision tree. As experimentation, we thought of having different decision tree with each decision tree based on subset of features and then each decision tree will participate in voting. Result obtained was further improvement on the score obtained from decision tree. We applied Decision tree and Random forest on the data which has been spell checked and the data where spell check was not used. The RMS results which were given by Kaggle have been tabulated below:

Method	Spell check = F	Spell checks = T
Decision Tree	0.6788	0.6718
Random Forest	0.5012	0.4974

VII. Individual Tasks

Prateek: I did the research about feature selection and how feature creation can affect the accuracy of predictive model. I also played part in data pre-processing which involved cleaning the data and transforming the raw vocabulary of search query,product title and product description into some meaningful features. Creation of new features was a challenge for application of both Decision Tree and Random Forest, thus each of us came up with ideas about overcoming those challenges and applying Decision tree.

Ritesh: The project started with research on the work done earlier for Recommendation systems and I along with Prateek did a part of it. Pre-processing was discussed among all group members and thus the part was done with equal contributions. New features were created for the decision tree algorithm and a lot of discussion was done for creation of the same. Decision tree and Random Forest was finally selected as the two algorithms we would use to classify the unlabelled data. We faced a lot of challenges as a team like cleaning the data and converting all the terms in search query to their root vocabulary.

Manish: At the very step, we were supposed to clean the data as there were many discrepancy in the data which could have hampered the result. So, to process the data I first had a look of data and saw what kind of error we can get if we run our algorithm on the current data. So, to remove those error I used regular expression to remove those error from data. Also, same word can occur multiple times with some prefix or suffix. It would have worsen the result. So, i wrote code to extract the root word without any prefix or suffix. To apply random forest on our data, we needed more features as each decision tree in ensemble method just use some subset of features. Then I ran random forest algorithm on constructed features to prepare model. On the built model, we ran the test data and got the predicted result.

All of us contributed equally in making this project a success.

References

[1] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schtze : An Introduction to Information Retrieval, Cambridge University Press Cambridge, England

[2] Tie-Yan Liu, Lead Researcher Microsoft Research Asia. Learning to Rank for Information Retrieval

[3] A Survey of Information Retrieval and Filtering Methods https://terpconnect.umd.edu/~oard/pdf/cstr3514.pdf

[4] Information Retrieval Wikipedia https://en.wikipedia.org/wiki/Information_retrieval

[5] OpenSource Connections http://opensourceconnections.com/